



Incorporating Cost-Benefit Analyses into Software Assurance Planning



Presentation to the
26th Annual Software Engineering Workshop
NASA/Goddard Space Flight Center (GSFC)
Software Engineering Laboratory (SEL)
and the IEEE Computer Society

Martin S. Feather, **Burton Sigal**, Steven L. Cornford
Jet Propulsion Laboratory
California Institute of Technology
&
Patrick Hutchinson
Wofford College, Spartanburg, SC

For further info contact: Martin.S.Feather@Jpl.Nasa.Gov
<http://eis.jpl.nasa.gov/~mfeather>



Software Assurance Planning



Software assurance is the planned and systematic set of activities that ensures that software processes and products conform to requirements, standards, and procedures.

Software Assurance Activities (inspections, tests, reviews,...)

- *Benefit*: reduce risk
- *Cost*: time, \$

Limited resources - must select activities judiciously

To do so, need means to *quantitatively* assess the cost/benefit of assurance activities applied to specific projects. This will:

- determine best use of limited resources
- identify alternatives (e.g., requirements to discard)
- be persuasive to developers and managers



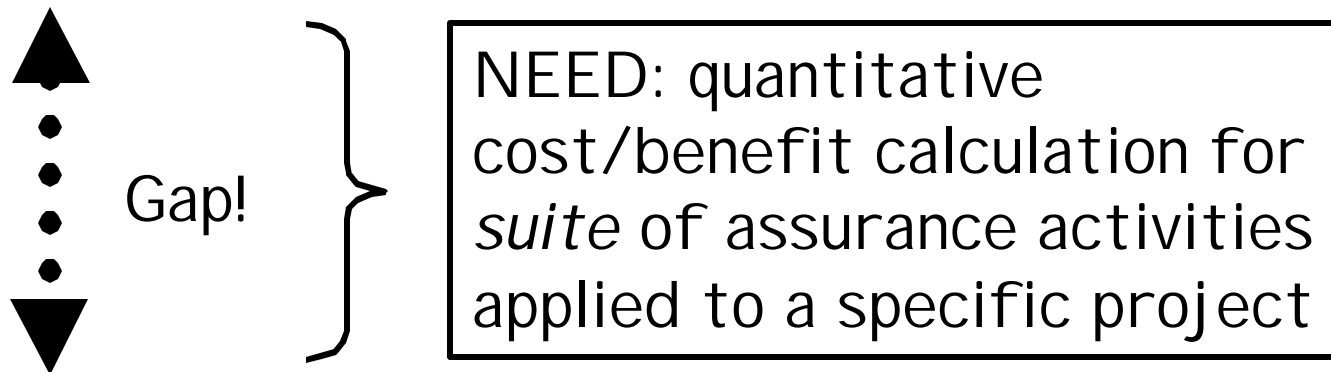
Cost/Benefit Reasoning for a *Suite* of Software Assurance Activities



Cost/benefit data & reasoning has been applied to:

Individual activities, e.g., Regression testing [Graves et al, 1998].

Pairwise comparisons, e.g., “Peer reviews are more effective than function testing for faults of omission and incorrect specification” [Basili & Boehm, 2000].



Lifecycle process improvement, e.g., Quality, productivity and estimation gains from CMM-like process improvement [McGarry et al, 1998].



Advanced Risk Reduction Tool (ARRT)



ARRT is inspired by, and based on
JPLer Steve Cornford's
Defect Detection and Prevention (DDP) process and tool

DDP *process* [Cornford et al, 2001]
supported by a custom *tool* [Feather et al, 2000]
for *quantitative* risk management.

ARRT is DDP augmented as follows:

- pre-populated with software assurance effectiveness data
- can be used in conjunction with NASA Glenn's Ask Pete tool
- has a sophisticated cost/benefit model



ARRT inherits DDP's model of risk mitigation



DDP utilizes three trees of key concepts:

- Requirements (what you want)

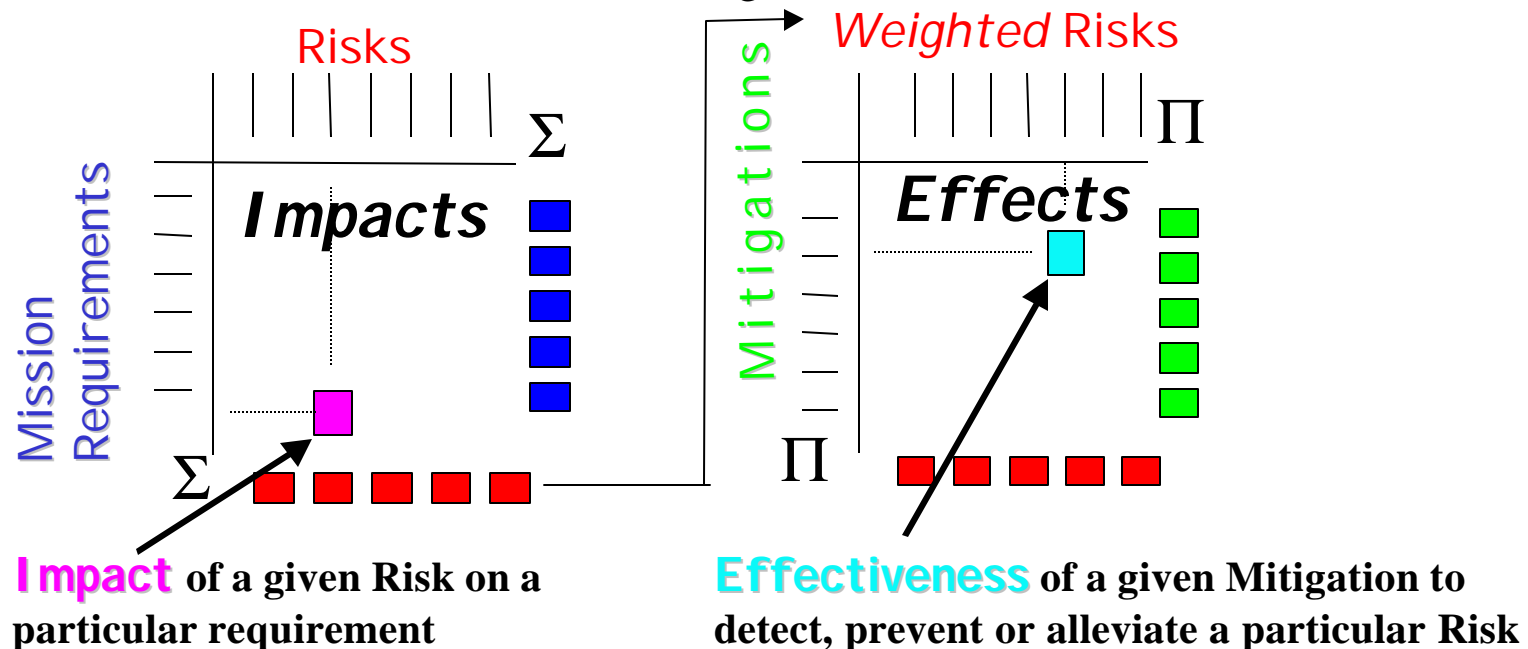
- Risks (what can get in the way of requirements)

- Mitigations (what can mitigate risk)

and two matrices that connect those concepts:

- Impacts (how much Requirement loss is caused by a Risk)

- Effectivenesses (how much a mitigation reduces a Risk)





ARRT's Quantitative Cost/Benefit Model



Risk mitigations subdivided into

Preventions – prevent problems from appearing in the first place

e.g., training programmers → fewer coding errors

cost = performing prevention

benefit = reduction of risk likelihood

Detections – detect problems so that they can be corrected

e.g., unit testing → detects internal coding errors

cost = performing detection +
performing the repair (cost depends on when!)

benefit = reduction of risk likelihood

Alleviations – applied to decrease the severity of problems

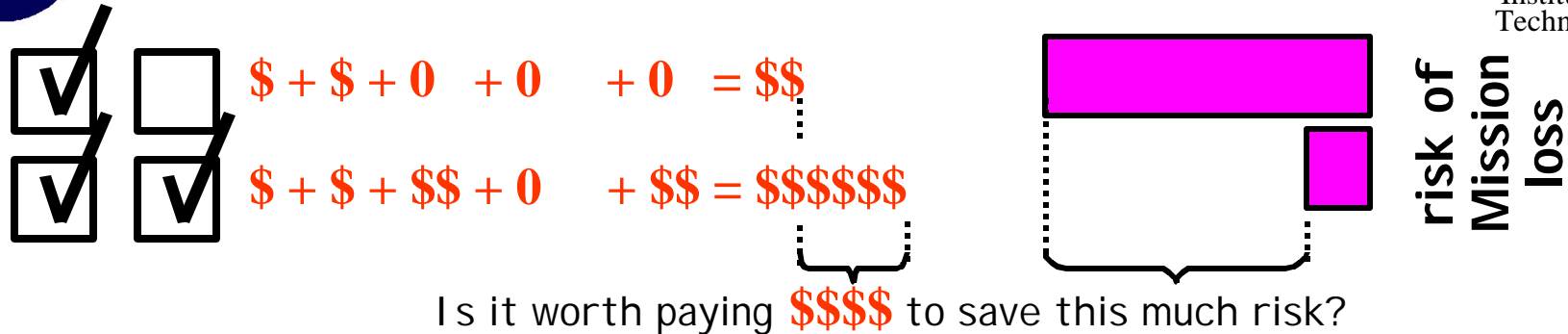
e.g., robust coding → tolerant of out-of-bound input values

cost = performing alleviation

benefit = reduction of risk severity



Return On Investment of Assurance



Return On Investment (ROI) calculation

ROI = benefit of risk reduction / cost of assurance

Conservative basis for ROI : benefit =

Mission cost * (Risk reduction due to Assurance)

- E.g., Mars Polar Lander + Mars Climate Orbiter missions cost = \$183,000,000

Aggressive basis for ROI : benefit =

(Value of attaining mission requirements) *
(Risk reduction due to Assurance)

- What is the value of discovering water on Mars?
- What is the value of returning a Mars sample to Earth?



ARRT's Quantitative Cost/Benefit Model



Cost/benefit computations in ARRT

- Automatic
- Handle *suite* of assurance activities
- Permit data to be changed if we know better than standard estimates
- Distinguish development phases (requirements, design, ...)
- Distinguish preventions, detections and alleviations
- Combine with underlying risk computation model



Software Estimation & Planning data: ARRT – Ask Pete collaboration



Ask Pete runs to gather project characteristics, make first cut at suggested selection of risk mitigations.

*Mitigation selection passed to **ARRT***

ARRT runs to allow user to assess risk, provide costs, **customize to project** (add/remove risks, refine effect values, etc.), tune selection accordingly.

Revised mitigation selection returned to Ask Pete

Ask Pete runs to generate final reports

see companion presentation in this workshop



Tim Kurtz, ε
Tim.Kurtz@grc.nasa.gov
SAIC/NASA Glenn Research Center
<http://tkurtz.grc.nasa.gov/pete>
Principal Investigator ε Martha Wetherholt



GOT RISK?



TOO MUCH – use ARRT to plan
how to reduce risk in a cost-effective manner.

TOO LITTLE – use ARRT to plan how to accept
more risk in exchange for reduced cost and
schedule, more functionality, etc.

JUST RIGHT – use ARRT to maintain a desired
risk profile through the lifetime of the project.

DON'T KNOW – use ARRT to assess risk status.

**"Risk as a Resource" – Dr. Michael Greenfield
[Greenfield, 1998]**



ARRT Availability



DDP:

.gov domains – available for immediate download
others - must apply for license

ARRT:

currently a variant compilation of DDP
in process of incorporating as a choice within DDP's
opening screen

go to:

<http://eis.jpl.nasa.gov/~mfeather>

& look for:

“Risk assessment and planning tools: DDP & ARRT”



References



- [Basili & Boehm, 2000] V. Basili & B. Boehm "CeBaSE: The Center for Empirically based Software Engineering" *NASA Goddard 25th Annual Software Engineering Workshop*, 2000.
- [Cornford et al, 2001] S.L. Cornford, M.S. Feather & K.A. Hicks. "DDP – A tool for life-cycle risk management", *IEEE Aerospace Conference*, Big Sky, Montana, Mar 2001, pp. 441-451.
- [Feather et al, 2000] M.S. Feather, S.L. Cornford & M. Gibbel. "Scalable Mechanisms for Requirements Interaction Management", *4th IEEE International Conference on Requirements Engineering*, Schaumburg, Illinois: 119-129, June 2000.



References



- [Graves et al, 1998] T. Graves, M. Harrold, J. Kim, A. Porter and G. Rothermel. "An Empirical Study of Regression Test Selection Techniques". *20th Int. Conference on Software Engineering*, 1998, pp. 267-273.
- [Greenfield, 1998] M.A. Greenfield "Risk Management 'Risk As A Resource' " *<http://www.hq.nasa.gov/office/codeq/risk/>*
- [Hoh & Roy, 2001] H. In & S. Roy "Visualization Issues for Software Requirements Negotiation" *25th Annual International Computer Software and Applications Conference*, Chicago, IL, Oct. 2001.
- [McGarry et al, 1998] F. McGarry, S. Burke & B. Decker. Measuring the impacts individual process maturity attributes have on software products., *5th International Software Metrics Symposium*, 1998, pp. 52-60



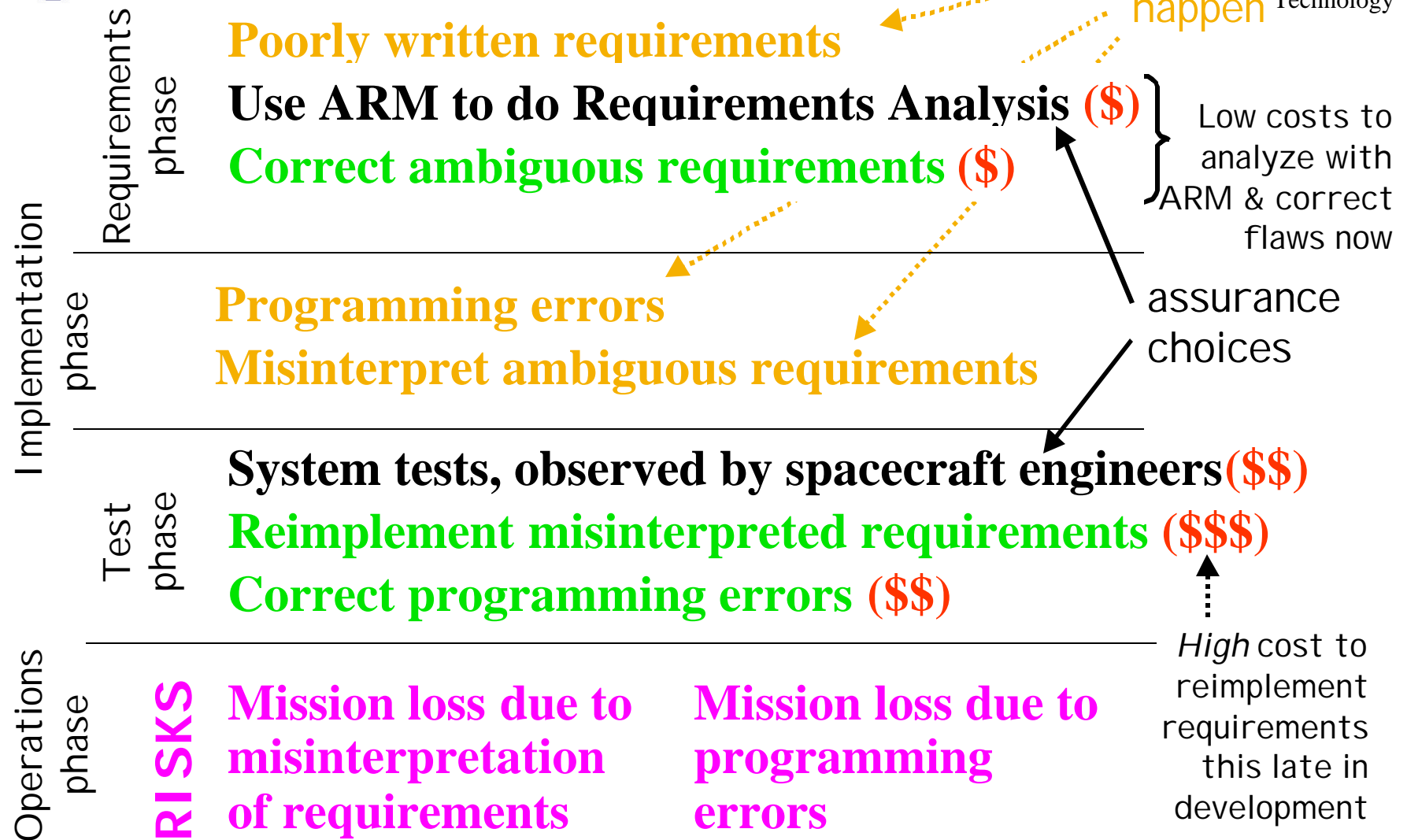
Backup Slides



California
Institute of
Technology

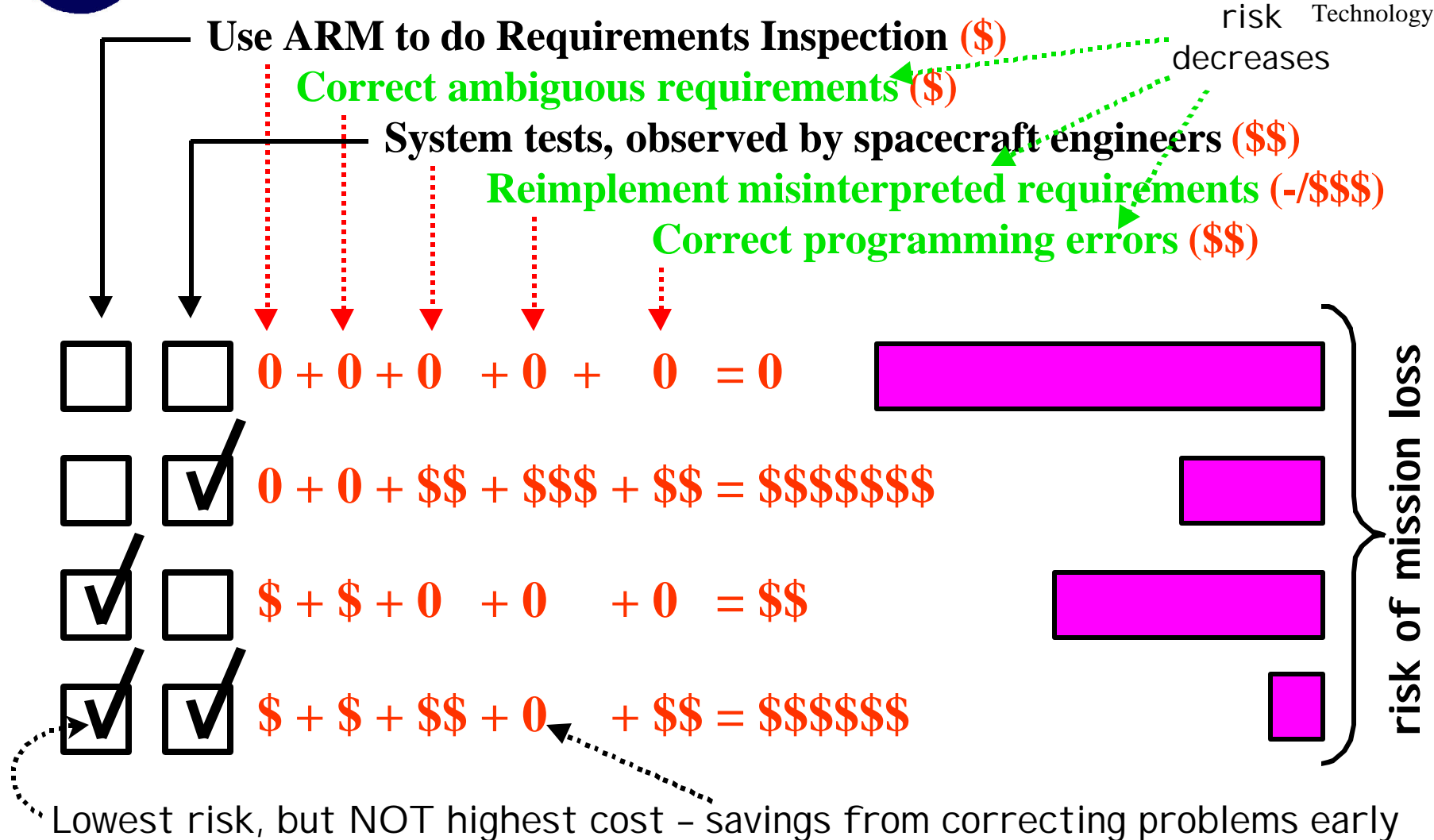
Cost/Benefit – Simple Scenario

mistakes
happen





Cost/Benefit – Simple Scenario (cont.)





Focused study data: Software Assessment Exercise



Steve Cornford, JPL + others

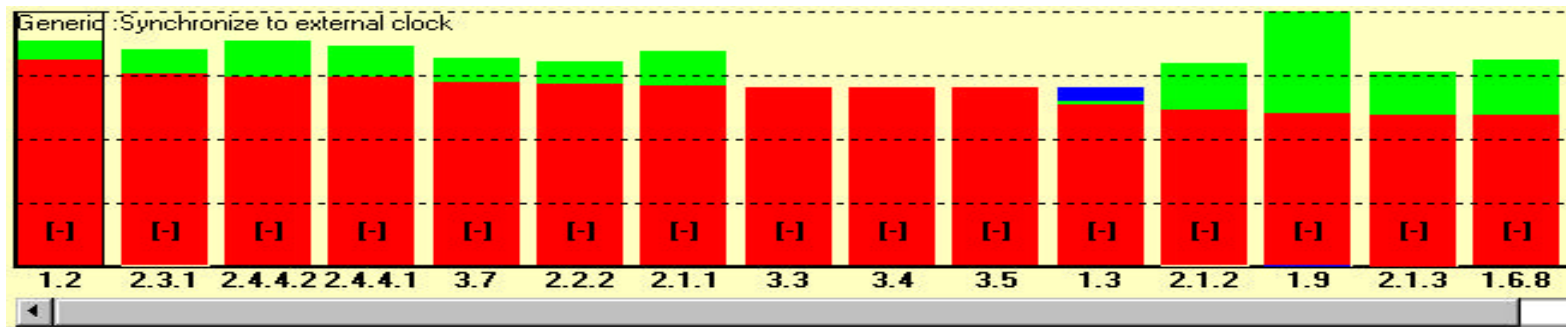
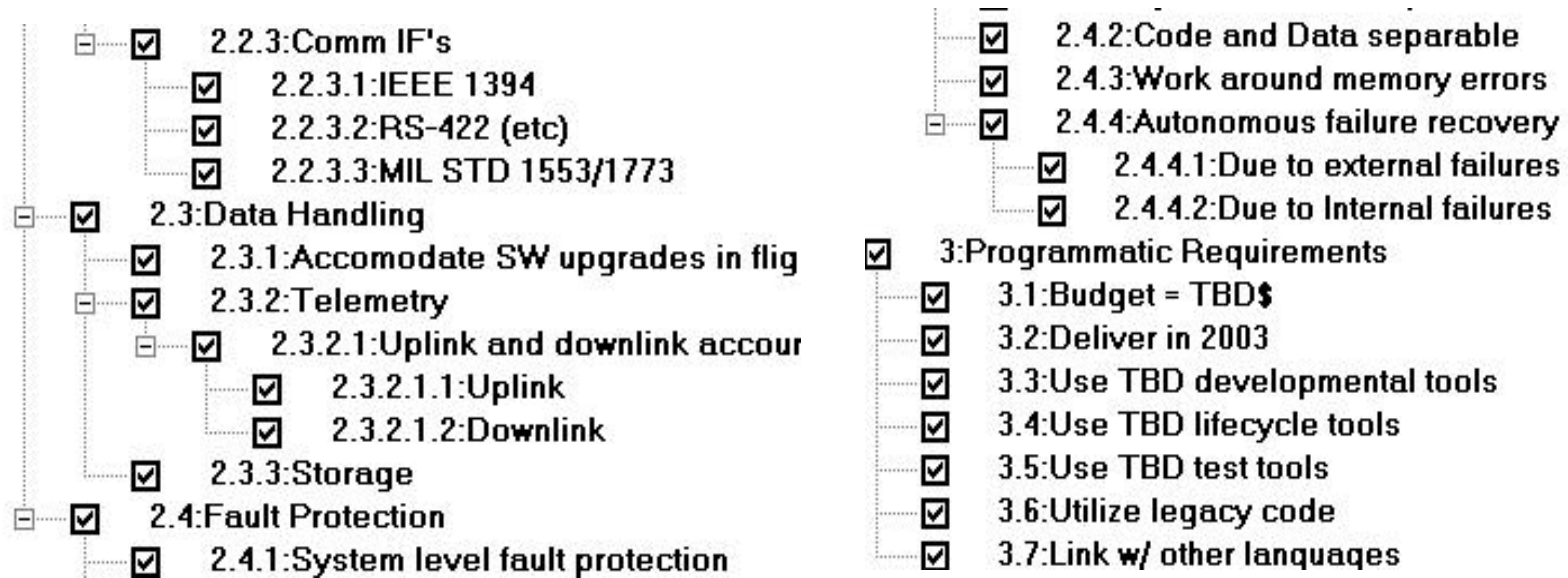
- Focus: code generation by [product name deliberately hidden]
 - Flight code of modest experiment
 - Flight code for future missions
- 15+ experts in 4 x 4-hour sessions, Sept 2000
 - [product] experts
 - Mission experts
 - Software experts (SQA, coders, ...)
- Large information set
 - 47 Requirements (unprioritized)
 - 76 Risks (near-term mission-specific & futuristic)
 - 303 Mitigations (pre-populated with large set)
 - 107 Impacts
 - 223 Effects



Software Assessment Exercise – extract



Portions of the Requirements tree and bar chart





Software Engineering Community Data



- Risks: Software Risk Taxonomy (SEI)
- Mitigations: two datasets:
 - JPL's Risk Balance Profile of SQA actions
 - Assurance activities from Ask Pete (NASA Glenn tool)
- Effects: cross-linkings of the above (Jim Kiper)
 - Expert's best estimates* of yes/no (Prof. J. Kiper)
 - Experts' 1000+ best estimates* of quantified effectiveness (Prof. J. Kiper & J. Eddingfield)

Note: Requirements are project specific

*** ARRT needs YOUR data!**



ARRT - Tim Menzies collaboration *in progress*



Prof. Tim Menzies, U. British Columbia

Benefits to ARRT of collaboration

- Optimization – automated search for (near) optimal mitigations suites
 - Least risk for given cost
 - Least cost for given risk
- Sensitivity analysis
 - On which data values do the results hinge?
 - Scrutinize these values further
 - Identify points of leverage (e.g., problematic requirements; make-or-break decisions)
- Retain human involvement
- Extend reasoning to more complex data
 - Interactions: mitigations that induce risk (e.g., code changes to correct one bug may introduce other bugs)
 - Ranges / distributions of values (e.g., [0.1 – 0.3])

tim@menzies.com





ARRT Heritage & Contributors



ARRT is inspired by, and based on
JPLer Steve Cornford's Defect Detection and Prevention (DDP)
and JPLer Tim Larson's Risk Balancing Profiles (RBP).

contributors (JPL)

John Kelly
Burt Sigal
James Eddingfield
Steve Cornford
Phil Daggett
Julia Dunphy
Roger Klemm

contributors

Jim Kiper (U. Miami, Ohio)
William Evanco (Drexel)
Steve Fickas (U. Oregon)
Martha Wetherholt (NASA Glenn)
Richard Hutchinson (Wofford, SC)

primary collaborators

Tim Menzies (U. British Columbia)
Tim Kurtz (NASA Glenn)
Hoh In (Texas A&M)

funding, management & guidance

NASA Code Q, NASA Goddard IV&V Facility
Siamak Yassini, Ken McGill, Marcus Fisher



ARRT/DDP Computations & Visualizations



Information is derived from user-provided data via built-in computations, e.g.,

- FM's cumulative impact = $\text{FM.Likelihood} * (\sum (R \in \text{Requirements}) R.\text{Weight} * \text{Impact}(R, \text{FM}))$

Information presented via cogent visualizations

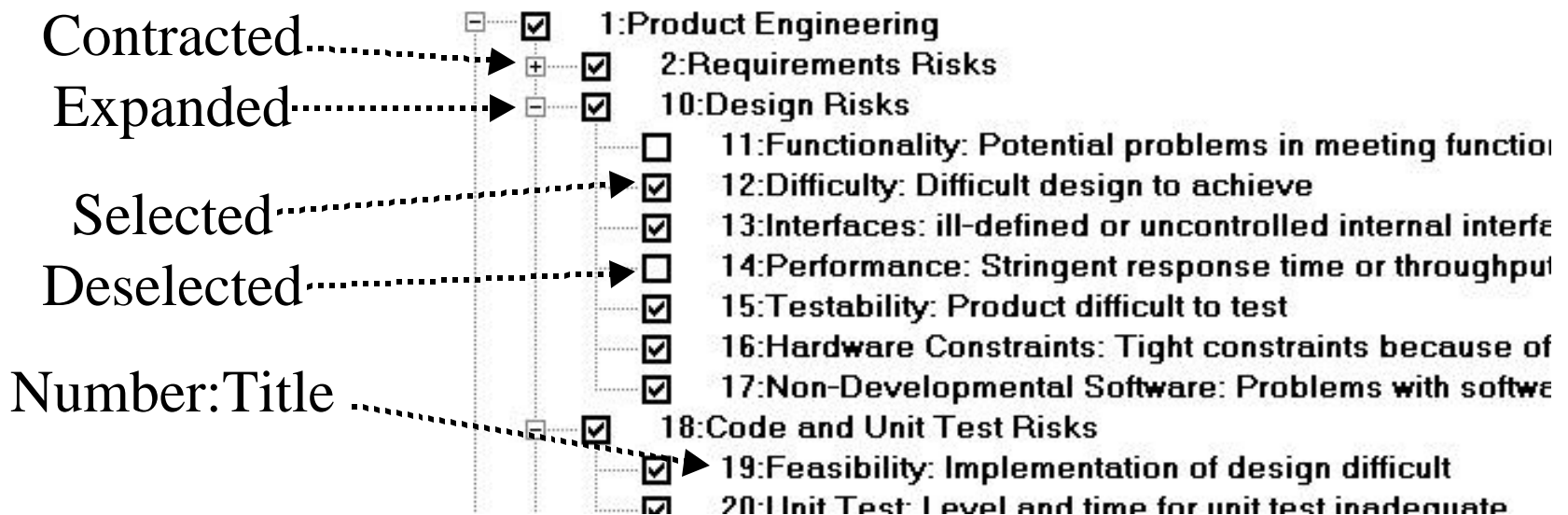
- Bar charts
- Risk Region chart
- Stem-and-leaf plots
- Detailed view of properties of individual element



ARRT/DDP Trees



Taxonomies of Software Requirements / Risks / Risk Mitigations



Autonumbering: *linear* 1,2,... or *tree* 1, 1.1, 1.2, 1.2.1, ...



ARRT/DDP Matrices



Effects (Mitigation x Risk)

		FMs	[-]Product Engineering					
		FMs	[-]Requirements Risks					
		FMs	Stabilit	Compli	Clarity	Validity	Feasib	Pre
PACTs	PACTs	FoMR	0.5	0.5	0.5	0.5	0.5	0.5
	Authori	7.95	0.1	0.1	0.1	0.1	0.1	0.3
	Identify	2.3						
	Mainte	0						
	Softwa	2.65						
	Implem	1.85	0.9	0.3	0.9	0.9	0.3	0.3
	Manag	0.15						
	Docum	1.65	0.3	0.9	0.9	0.1	0.3	0.3
	Peer	2.8	0.9	0.9	0.9	0.9	0.9	0.9

numbers
supplied by
experts and/or
based on
accumulated
metrics

proportion of
Risk reduced
by Mitigation

Impacts (Requirement x Risk):

proportion of Requirement loss if Risk occurs

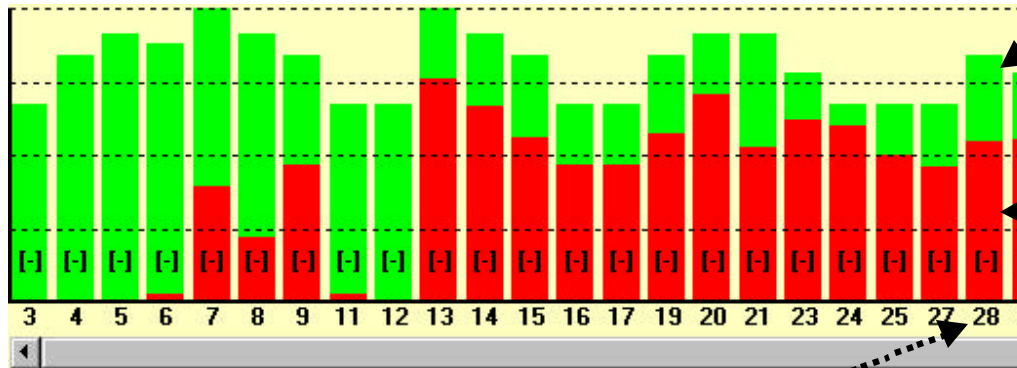


ARRT/DDP Visualizations - Bar Charts



Risks bar chart

Unsorted – order matches leaf elements in Risk tree

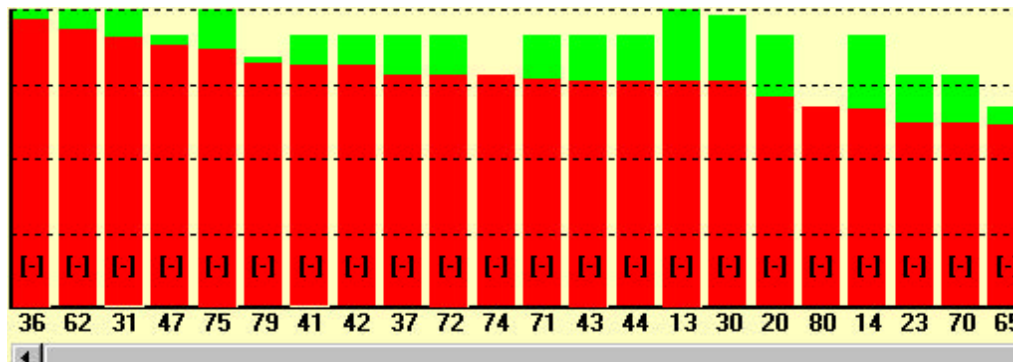


Item number in tree

Green: of this Risk's total Impact on Requirements, that *saved* by Mitigations

Red: of this Risks's total Impact on Requirements, that *remaining* despite Mitigations

Sorted – in decreasing order of remaining risk



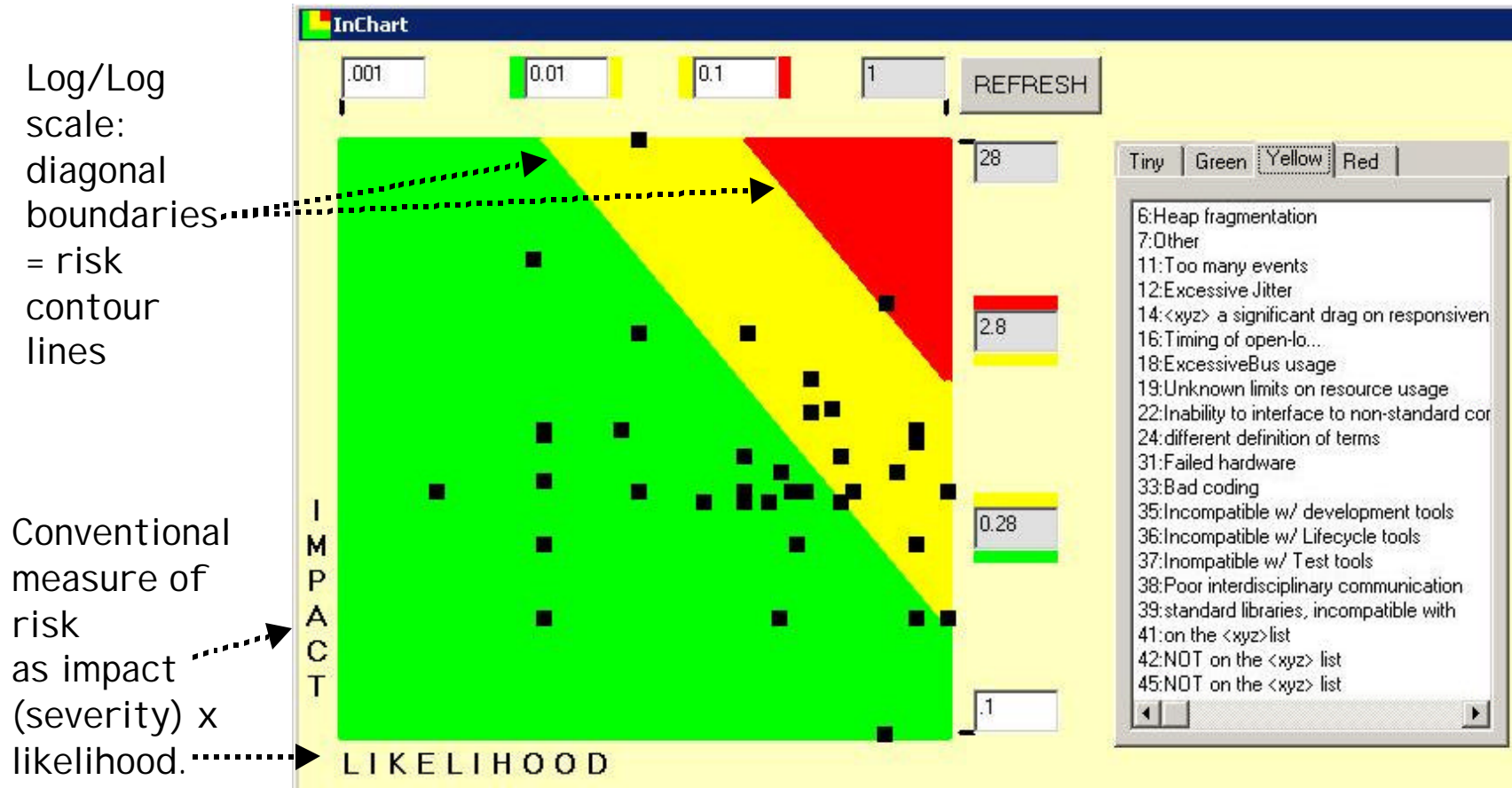
Requirements bar chart – how much each is impacted

Mitigations bar chart – how much impact each is saving



ARRT/DDP Visualizations – Risk Region – “InChart”

User defines risk levels demarking red/yellow/green/(tiny) risk regions

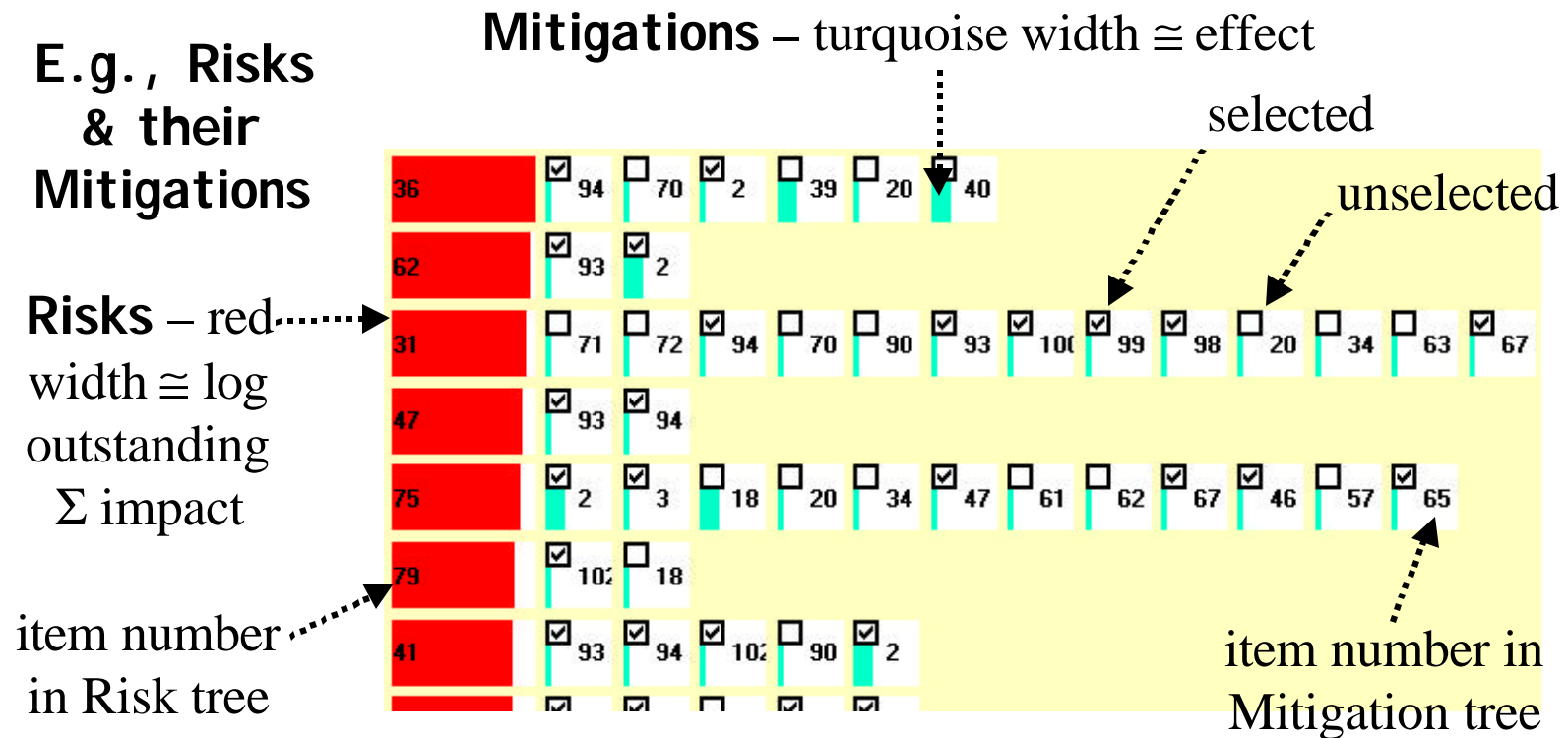




ARRT/DDP Visualizations – stem-and-leaf(*) charts



Compact visualization of DDP's sparse matrices



(*) Tufte attributes these to John W. Tukey, “Some Graphical and Semigraphic Displays”
Their usage was introduced into RBP by D. Howard, extended further by us in DDP.